MEMORANDUM
RM-4287-PR
OCTOBER 1964

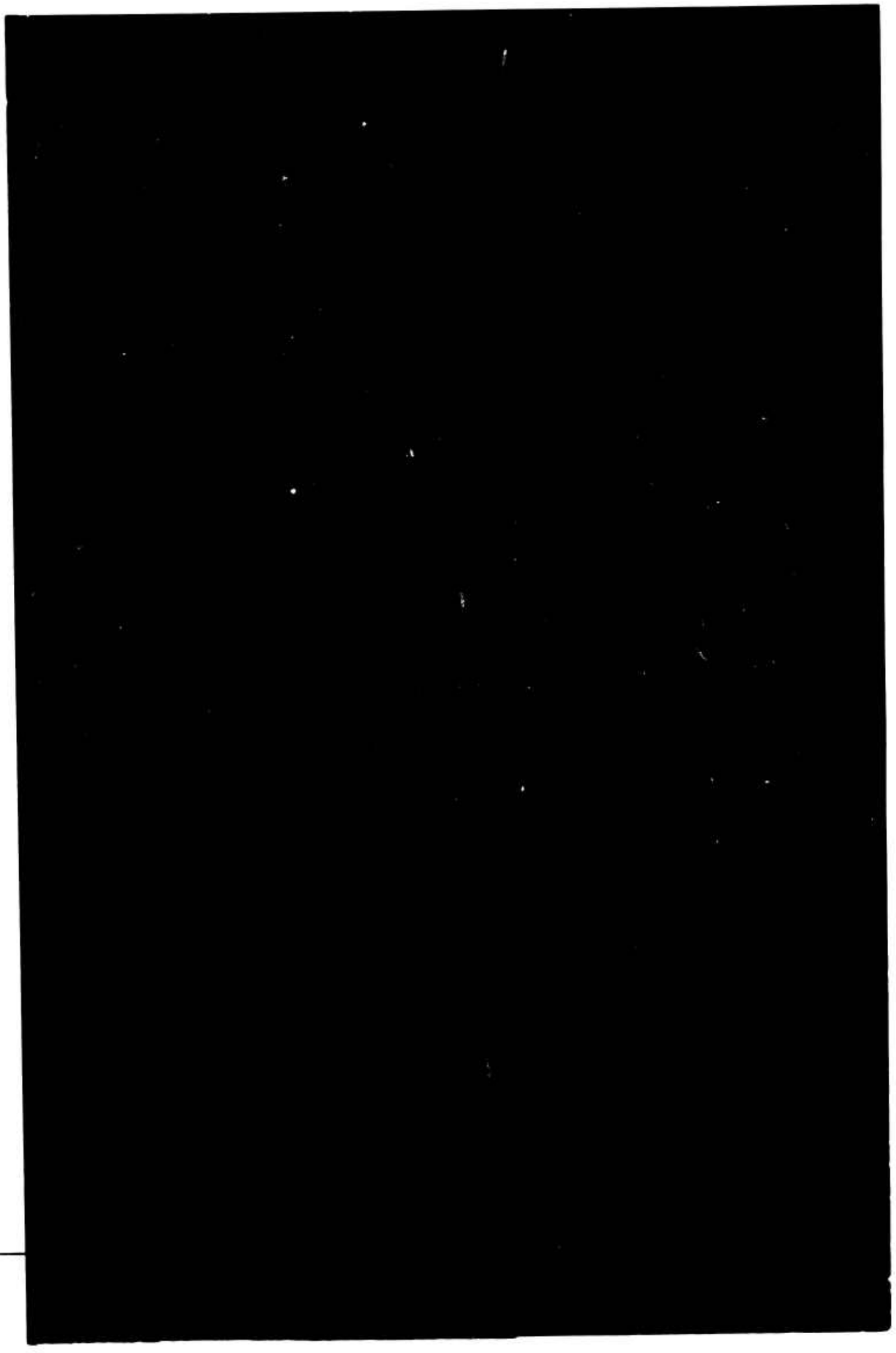
16 p

ON THE COMPUTATIONAL SOLUTION OF A CLASS OF FUNCTIONAL DIFFERENTIAL EQUATIONS

Richard Bellman and K. L. Cooke

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA



MEMORANDUM
RM-4287-PR
OCTOBER 1964

ON THE COMPUTATIONAL SOLUTION OF A CLASS OF FUNCTIONAL DIFFERENTIAL EQUATIONS

Richard Bellman and K. L. Cooke

This research is sponsored by the United States Air Force under Project RAND—Contract No. AF 49(638)-700 monitored by the Directorate of Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF. Views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of the United States Air Force.

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA • 90406

PREFACE

Part of the Project RAND research program consists of basic supporting studies in mathematics. This Memorandum presents a method for the computational solution of a type of differential equation that frequently arises in the construction of mathematical models in a variety of fields, including engineering and biophysics.

SUMMARY

Functional differential equations of the form

(1) $$u'(t) = g(t,u(t),u(h(t))),$$

and, more generally, of the form

(2) $$u'(t) = g(t,u(t),u(h(u,t))),$$

arise in the construction of realistic models in a number of fields, ranging from electromagnetic theory and control theory to respiratory theory and neurophysiology. The analytic aspects are quite complex, and numerical solution is anything but routine, even with the aid of a digital computer. In this paper, we wish to extend a method for the computational treatment of differential–difference equations to cover equations of the form given in (1). Equations of the type appearing in (2) can then be treated by means of successive approximations.

CONTENTS

BLANK PAGE

ON THE COMPUTATIONAL SOLUTION OF A CLASS OF FUNCTIONAL DIFFERENTIAL EQUATIONS

1. INTRODUCTION

Functional differential equations of the form

$$u'(t) = g(t,u(t),u(h(t))), \tag{1.1}$$

and, more generally, of the form

$$u'(t) = g(t,u(t),u(h(u,t))), \tag{1.2}$$

arise in the construction of realistic models in a number of fields, ranging from electromagnetic theory and control theory to respiratory theory and neurophysiology. The analytic aspects are quite complex (see [1], [2]), and numerical solution is anything but routine, even with the aid of a digital computer. In this paper, we wish to extend a method given in [3,4,5] for the computational treatment of differential-difference equations, to cover equations of the form given in (1.1). Equations of the type appearing in (1.2) can then be treated by means of successive approximations.

2. THE EQUATION $u'(t) = g(u(t),u(h(t)))$

Let us suppose that $h(t) \leq t$, for $t \geq 0$, so that the future is determined by the past. An objection to

using straightforward techniques for the numerical solution of

$$u'(t) = g(u(t),u(h(t))), \tag{2.1}$$

with appropriate initial conditions, lies in the amount of rapid-access storage required for large t. If this equation is merely one of a number of equations which arise in the description of a large, complex system, we cannot afford to use up rapid-access storage.

Let us suppose that $h'(t) > 0$ for $t \geq 0$, and let the inverse function $h^{-1}(t)$ be denoted by $H(t)$. Suppose further that $u(t)$ is known in some initial interval $[0,t_1]$, where $t_1 = H(0)$, and let the sequence $\{t_n\}$ be defined recursively by

$$t_n = H(t_{n-1}), \quad n = 2,3,.... \tag{2.2}$$

Let $H^{(n)}(t)$ denote the n-th iterate of $H(t)$, $H^{(n)} = H(H^{(n-1)})$, $n = 2,3,....$ We observe that the function $H(t)$ maps the interval $[t_{n-1},t_n]$ onto $[t_n,t_{n+1}]$ in one-to-one fashion, $n = 1,2,...,$ and $H^{(k)}(t)$ maps $[t_{n-1},t_n]$ onto $[t_{n-1+k},t_{n+k}]$.

Consider the function

$$u_n(s) = u(H^{(n)}(s)), \quad n = 0,1,2,..., \tag{2.3}$$

where s is restricted to the interval $[0,t_1]$, and $H^{(0)}(s) = s$. Thus the values of $u_n(s)$, for $0 \le s \le t_1$, are the values of $u(t)$ for $t_n \le t \le t_{n+1}$. We have

$$(2.4) \qquad u_n'(s) = [\frac{d}{ds} H^{(n)}(s)]u'(H^{(n)}(s)),$$

and the derivative of $H^{(n)}(s)$ can easily be evaluated recursively by the formula

$$(2.5) \qquad \frac{d}{ds} H^{(n)}(s) = H'(H^{(n-1)}(s)) \frac{d}{ds} H^{(n-1)}(s).$$

Now we set $t = H^{(n)}(s)$, where $0 \le s \le t_1$. Then from the equation in (2.1) we get

$$(2.6) \qquad u'(H^{(n)}(s)) = g\{u(H^{(n)}(s)),u(H^{(n-1)}(s))\}$$

$$= g(u_n(s),u_{n-1}(s)), \quad n = 1,2,....$$

Referring to (2.4), we see that (2.6) may be written

$$(2.7) \qquad u_n'(s) = [\frac{d}{ds} H^{(n)}(s)]g(u_n(s),u_{n-1}(s)), \quad n = 1,2,....$$

Thus (2.1) has been replaced by a system of ordinary differential equations where s now ranges over a fixed interval $[0,t_1]$. This is important from the computational point of view.

3. NUMERICAL SOLUTION

The computational solution of this system is not routine, since we do not possess the requisite initial values $\{u_n(0)\}$ for $n \geq 2$. Consequently, we proceed as in [3]. The equation

$$(3.1) \qquad u_1'(s) = H'(s)g(u_1(s),u_0(s)), \quad u_1(0) = u_0(t_1)$$

gives us the value $u_2(0) = u_1(t_1)$. We now consider the simultaneous equations

$$(3.2) \qquad u_1'(s) = H'(s)g(u_1(s),u_0(s)), \quad u_1(0) = u_0(t_1),$$

$$u_2'(s) = [H^{(2)}(s)]'g(u_2(s),u_1(s)), \quad u_2(0) = u_1(t_1),$$

and solve these to obtain $u_3(0) = u_2(t_1)$.

Continuing in this way, we require no storage of functions, at the expense of being required to solve successively larger systems of equations.

4. ANOTHER METHOD

Reduction of equation (2.1) to a system of ordinary differential equations over a fixed interval can also be achieved by introducing new independent and dependent variables in a manner attributed originally to Laplace (see [1], page 84). It is also closely related to some

ideas of Abel concerning the iteration of functions. We begin by introducing a new independent variable x by the equations

$$(4.1) \qquad t = p(x), \quad h(t) = p(x - \xi),$$

where ξ is an arbitrary fixed constant. The function p must be chosen so that

$$(4.2) \qquad h(p(x)) = p(x - \xi),$$

or

$$(4.3) \qquad p(x) = H(p(x - \xi)),$$

and should be monotone and continuous, in order that the relation $t = p(x)$ be solvable for $x = p^{-1}(t)$. We note that if p is any monotone function on $0 \leq x \leq \xi$ with the property that

$$(4.4) \qquad h(p(\xi)) = p(0),$$

then because H is continuous and monotone. it will follow that $p(x)$ is continuous and monotone on $0 \leq x \leq 2\xi$, and by iteration of (4.3), continuous and monotone on $0 \leq x$.

Let

$$v(x) = u(t) = u(p(x)). \tag{4.5}$$

It then follows that

$$v'(x) = u'(t)p'(x) = u'(p(x))p'(x).$$

Since (2.1) yields

$$u'(p(x)) = g\{u(p(x)),u(h(p(x)))\}$$

$$= g\{u(p(x)),u(p(x - \xi))\},$$

and since $u(p(x)) = v(x)$ and $u(p(x - \xi)) = v(x - \xi)$, we see that

$$v'(x) = p'(x)g(v(x),v(x - \xi)). \tag{4.6}$$

Thus the introduction of new independent and dependent variables x and v by means of (4.1) and (4.5) leads to the replacement of (2.1) by the pair of equations (4.2) and (4.6), one of which is a difference equation and the other a differential-difference equation, both with a fixed lag ξ. It is not hard to see that if p and v are continuous solutions of (4.2) and (4.6), then (4.1) and (4.5) define a solution $u(t)$ of (2.1).

The equations (4.2) and (4.6) can now be reduced to a system of equations on the interval $[0,\xi]$ by the technique of [4]. That is, we define

$$(4.7) \qquad p_n(x) = p(x + n\xi), \qquad (0 \leq x \leq \xi;\ n = 0,1,\ldots)$$
$$v_n(x) = v(x + n\xi).$$

Here by $p_0(x)$ and $v_0(x)$ we accordingly mean the initial values of $p(x)$ and $v(x)$ on $0 \leq x \leq \xi$. With these definitions, equations (4.3) and (4.6) finally take the form

$$(4.8) \qquad p_n(x) = H(p_{n-1}(x)), \quad (0 \leq x \leq \xi;\ n = 1,2,\ldots)$$

$$(4.9) \qquad v_n'(x) = p_n'(x)g(v_n(x),v_{n-1}(x)).$$

The solution of (4.8) can now be carried out by iteration, and the solution of (4.9) by the method sketched in Sec. 3. The function $p_n'(x)$ can be evaluated recursively from the formula

$$p_n'(x) = H'(p_{n-1}(x))p_{n-1}'(x).$$

The similarity of (4.8) to the relation $H^{(n)}(s) = H(H^{(n-1)}(s))$, and of (4.9) to (2.7), is

apparent. Indeed, if we choose p_0 so that $p_0(x) = x$, then from (4.4) we obtain $h(\xi) = 0$, and ξ is the same as the t_1 defined in Sec. 2. Moreover, we then obtain $p_n(x) = H^{(n)}(x)$, and since

$$v_n(x) = v(x + n\xi) = u(p(x + n\xi)) = u(p_n(x)),$$

the function $v_n(x)$ is identical to the function $u_n(s)$ of Sec. 2.

On the other hand, the formulation in this section seemingly allows some extra latitude, since the choice of $p_0(x)$ is largely arbitrary. In practice, however, it is unlikely that any advantage can be derived from this.

5. AN EXAMPLE

We wish to consider a particular case, in order to illustrate a possible pitfall in the application of the techniques given above. Suppose the equation in question is

$$(5.1) \qquad u'(t) = g(u(t), u(\tfrac{t}{2} - 1)),$$

so that $h(t) = t/2 - 1$, $H(s) = 2(s + 1)$. Then $t_1 = 2$, $t_2 = 6$, $t_3 = 14$, and so on, and

$$H^{(n)}(s) = 2^n s + t_n.$$

Equation (2.7) becomes

$$u_n'(s) = 2^n g(u_n(s), u_{n-1}(s)).$$

Now suppose we carry out the solution of this system, determining $u_n(s)$ at equally spaced points $s = 0, \delta, \ldots, N\delta$, where $N\delta = t_1$. Referring to (2.3), we see that we obtain in this way the values of $u(t)$ at the points $t_n, t_n + 2^n\delta, t_n + 2^n 2\delta, \ldots, t_{n+1}$. Since the length of the interval $[t_n, t_{n+1}]$ doubles when n increases by one, whereas the number of points at which we know $u(t)$ is unchanged, it is evident that our computed values provide less and less information about $u(t)$ the larger t becomes.

Two ways of overcoming this difficulty suggest themselves. One is simply to use a very small value of δ in the first place—but this exacts a penalty in storage and computing time. The second is to combine the above scheme with an interpolation process. For the equation in (5.1), this could take the following form. Starting with $u_0(s)$, tabulated at $0, \delta, \ldots, N\delta$, solve the differential equation to obtain $u_1(s)$, and consequently $u(t)$ at N points on the interval $[t_1, t_2]$. By interpolation, subtabulate $u(t)$ at N additional points. This yields $u_1(s)$ at $2N$ points, say $s = 0, \delta/2, \delta, 3\delta/2, \ldots, N\delta$. Continuing in this way, we

obtain $u(t)$ at equally spaced points over any interval. It seems likely that the insertion of an interpolation at each step of the process, or whenever the computed points become too thinly distributed, will improve the accuracy of the results.

In many applications of interest, fortunately, $h(t)$ is such that $h(t) \sim t - b$, b a constant, as $t \to \infty$. Hence, the foregoing difficulty does not arise.

6. SUCCESSIVE APPROXIMATIONS

A minor modification enables us to treat the more general equation of (1.1). The equation of (1.2), which arises from some realistic models of respiratory processes, can be treated by means of successive approximations.

REFERENCES

1. Bellman, R., and K. L. Cooke, Differential-difference Equations, Academic Press Inc., New York, 1963.

2. Myskis, A. D., Linear Differential Equations with a Delayed Argument, Moscow, 1951.

3. Bellman, R., and B. Kotkin, "On the Numerical Solution of a Differential-difference Equation Arising in Analytic Number Theory," Math. of Comp., Vol. 16, 1962, pp. 473-475.

4. Bellman, R., "On the Computational Solution of Differential-difference Equations," J. Math. Anal. Appl., Vol. 2, 1961, pp. 108-110.

5. ———, "Successive Approximations and Computer Storage Problems in Ordinary Differential Equations," Comm. Assoc. Comput. Machinery, Vol. 4, 1961, pp. 222-223.